b) [generate] generating a pointer for each of said regions [pointers], each of said pointers associating [one of said regions] its respective region with one of said textures[, said regions comprising pixels]; and

B11 conc

c) [store the map] storing the bitmap and the pointers in a memory coupled to the microprocessor.

Remarks

Applicant thanks the examiner for the telephone interview of May 23, 1997. Applicant's responses are set forth below, in each case following a quotation (indented and in bold face small type) of the examiner's comment to which it relates.

> **2. Applicant's arguments filed February 11, 1997 have been fully considered but they are not persuasive. . . .**
>
> **4. Claims 1-3 and 34 are rejected under 35 U.S.C. 103 as being unpatentable over Leach. . . .**

Claim 1 has been amended to require "compressing a digital image" "prior to a time when the image is to be displayed" by storing a "bitmap representing boundaries separating regions in said image", where "said regions comprise pixels of said image" and storing "a pointer for each of said regions" that associates "its respective region with one of said textures." Claim 34 requires a "memory being configured to cause [a] microprocessor to compress a digital image . . . prior to a time when the image is to be displayed" by storing a "bitmap representing boundaries

separating regions in said image" and storing "a pointer for each of said regions" that associates "its respective region with one of said textures." One advantage of the claimed invention is that a bitmapped image (e.g., a sprite image) having distinct yet coupled regions of different textures (e.g., patterns of color) can be efficiently compressed in size for storage, and quickly decompressed for display.

By contrast, Leach does not describe or suggest compressing his sprite images; each of his sprites is simply fetched from memory as a horizontal line of bits for each vertical position of the sprite. (Col. 6:67 - Col. 7:2). Each sprite located on a particular display line has its horizontal line of bits placed in a sprite register 100, using a sprite horizontal pointer 82, pattern register 81, color register 80, and sprite coincidence selection logic 70. The sprite pattern register 81 then shifts bits out serially to the video RAM. [Col. 8:32-40]. While Leach employs color codes, such codes appear to be applied bit-by-bit for each sprite bitmap, and are not referenced by one pointer for a larger region of pixels. While Leach does mention a "pointer" (Col. 8:36-40), it is for a sprite horizontal pointer that merely contains the sprite's horizontal position, and does not associate a particular texture with a particular sprite region. Leach does not even hint at generating a bitmap representing boundaries separating regions, let alone generating pointers, each pointer associating its

respective region with one of the textures, as in claims 1 and 34.

> **5. Claims 1-11, 14-19, 22-23, 27-28, 31, 33, and 34 are rejected under 35 U.S.C. 103 as being unpatentable over Murata et al. . . .**
>
> **6. Claims 12-13, 20-21, and 24-26 are rejected under 35 U.S.C. 103 as being unpatentable over Murata et al. in view of Snyder et al. or Golin et al. . . .**
>
> **7. Claims 29 and 30 are rejected under 35 U.S.C. 103 as being unpatentable over Murata et al. in view of Foley et al. . . .**
>
> **8. Claim 32 is rejected under 35 U.S.C. 103 as being unpatentable over Murata et al. in view of Snyder et al. . . .**

Claim 14, similar to claim 1, requires "generating a bitmap representing boundary pixels of a first one of said textures separating said regions in said image". Claims 15, 22, 31, 33, and 34 require "a bitmap representing boundaries separating regions" with "said boundaries comprising pixels of said image" and "at least one of said regions comprising pixels of said image". Murata does not describe or suggest compressing an image as a <u>bitmap</u> of boundaries separating regions and pointers that associate textures with the regions. Murata first generates <u>vector</u> polygonal shapes, transforms the shapes (e.g., to exhibit perspective) and their associated textures, and then applies the transformed textures to the surfaces of these <u>vector</u> polygonal shapes. Murata applies textures to the surfaces generated by a transformed polygon, first to the calculated vertices of the polygon, and then to interpolated points between those vertices. [Col. 3:44 - Col. 4:7, Col. 16:40 - Col. 17:16]:

> The mapping of the texture data to the respective dots of the polygon is performed by specifying only texture coordinates

corresponding to each vertex in the polygon. The mapping of the texture data to dots other than the vertices may be made by interpolating the texture coordinates for each vertex to determine texture coordinates to be applied to the other dots and by reading out texture data through the interpolated texture coordinates. [Col. 17:9-16] (emphasis added).

In Murata, boundary maps are not compressed and stored as bitmaps. The passage at col. 18:20-23, cited by the Examiner, refers to the final generated bitmap of the full video image created by Murata's image synthesis system, not to a boundary map.

Claims 2-11, 12-13, 20-21, 24-26, 29-30, 32, and 34 are patentable for at least the same reasons as the claims from which they depend.

Applicant asks that all claims now be allowed.

Apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 4 June 1997

Hans R. Troesch
Rep. No. 36,950

Wayne P. Sobon
Reg. No. 32,438

Fish & Richardson P.C.
2200 Sand Hill Road, Suite 100
Menlo Park, CA 94025

Telephone: 415/322-5070
Facsimile: 415/854-0875

69501.p11